DATA SCIENCE ASSIGNMENT

DATA VISUALISATION

(2019A7PS0231U - HITESH GARG , 2019A7PS0253U - ROHAN KATOCH)

Code:

#importing libraries and packages

import numpy as num

import pandas as pds

import matplotlib.pyplot as mpl

import seaborn as sbn

#DATASET

#loading the dataset

zom\_og=pds.read\_csv("zomato.csv")

zom\_og.head()

#summary of the data

zom\_og.info()

#DATA CLEANING

#deleting unwanted columns such as dishliked,url,phone

zom\_new=zom\_og.drop(['url','dish\_liked','phone'],axis=1)

#removing the duplicate values

zom\_new.duplicated().sum()

zom\_new.drop\_duplicates(inplace=True)

#removing the nan and null values from our data

zom\_new.isnull().sum()

zom\_new.dropna(how='any',inplace=True)

zom\_new.info()

#checking the columns left

zom\_new.columns

#changing the column names

zom\_new = zom\_new.rename(columns={'approximately\_cost\_2\_people':'cost','listed\_in(type)':'type',

'listed\_in(city)':'city'})

zom\_new.columns

#changing the datatype for cost

zom\_new['cost'] = zom\_new['cost'].astype(str) # Changing the cost to string

zom\_new['cost'] = zom\_new['cost'].apply(lambda a: a.replace(',','.')) #lambda function to replace , to . from cost

zom\_new['cost'] = zom\_new['cost'].astype(float) # Changing the cost to Float

zom\_new.info()

#displaying and checking the rate column of data

zom\_new['rate'].unique()

#removing '/5' from rates

zom\_new = zom\_new.loc[zom\_new.rate !='NEW']

zom\_new = zom\_new.loc[zom\_new.rate !='-'].reset\_index(drop=True)

remove\_slash = lambda a: a.replace('/5', '') if type(a) == num.str else a

zom\_new.rate = zom\_new.rate.apply(remove\_slash).str.strip().astype('float')

zom\_new['rate'].head()

# Adjusting column names

zom\_new.name = zom\_new.name.apply(lambda a:a.title())

zom\_new.online\_order.replace(('Yes','No'),(True, False),inplace=True)

zom\_new.book\_table.replace(('Yes','No'),(True, False),inplace=True)

zom\_new.cost.unique()

#Encoding variables

def enc(zom\_new):

for column in zom\_new.columns[~zom\_new.columns.isin(['rate', 'cost', 'votes'])]:

zom\_new[column] = zom\_new[column].factorize()[0]

return zom\_new

zom\_en = enc(zom\_new.copy())

#correlation matrix

corr = zom\_en.corr(method='kendall')

mpl.figure(figsize=(12,8))

sbn.heatmap(corr, annot=True)

mpl.title('Correlation matrix',fontsize=18)

mpl.savefig("correlation.png")

zom\_en.columns

#Data visualization part

fig = mpl.figure(figsize=(20,8))

loc = sbn.countplot(x="location",data=zom\_og, palette = "Set1")

loc.set\_xticklabels(loc.get\_xticklabels(), rotation=90, ha="right")

mpl.ylabel("frequency(f)",size=12)

mpl.xlabel("Locality",size=12)

mpl.title('Restaurants in a location',fontsize = 18,pad=20)

mpl.savefig("Restaurants in a location.png")

fig = mpl.figure(figsize=(12,8))

rest = sbn.countplot(x="rest\_type",data=zom\_og, palette = "Set1")

rest.set\_xticklabels(rest.get\_xticklabels(), rotation=90, ha="right")

mpl.ylabel("frequency(f)",size=12)

mpl.xlabel("Type of restaurant",size=12)

mpl.title('Type of rest.',fontsize = 18 ,pad=20)

mpl.savefig('Type of restaurant.png')

mpl.figure(figsize=(12,8))

chains=zom\_og['name'].value\_counts()[:20]

sbn.barplot(x=chains,y=chains.index,palette='Set1')

mpl.title("Popular restaurants in Bangalore",fontsize=18,pad=20)

mpl.xlabel(" Outlets ",size=12)

mpl.savefig('Famous restaurant chains in Bangalore.png')

sbn.countplot(zom\_new['online\_order'])

fig = mpl.gcf()

fig.set\_size\_inches(12,12)

mpl.title('onlinedelivery or not',fontsize=18)

mpl.savefig("online.png")

sbn.countplot(zom\_new['book\_table'])

fig = mpl.gcf()

fig.set\_size\_inches(12,12)

mpl.savefig("Book\_Table.png")

mpl.title('Booking Table in a Restaurant (allowed or not)',fontsize=18)

mpl.rcParams['figure.figsize'] = (12, 9)

Y = pds.crosstab(zom\_new['rate'], zom\_new['book\_table'])

Y.div(Y.sum(1).astype(float), axis = 0).plot(kind = 'bar', stacked = True,color=['blue','red'])

mpl.title('table booking vs rate',fontsize = 18)

mpl.legend(loc="upper right")

mpl.savefig("Table\_Booking\_Rate.png")

mpl.show()

sbn.countplot(zom\_new['city'])

sbn.countplot(zom\_new['city']).set\_xticklabels(sbn.countplot(zom\_new['city']).get\_xticklabels(), rotation=90, ha="right")

fig = mpl.gcf()

fig.set\_size\_inches(12,12)

mpl.savefig("Location.png")

mpl.title('Location',fontsize=18)

loc\_plt=pds.crosstab(zom\_new['rate'],zom\_new['city'])

loc\_plt.plot(kind='bar',stacked=True);

mpl.title('Location - Rating',fontsize=18)

mpl.ylabel('Location',fontsize=12)

mpl.xlabel('Rating',fontsize=12)

mpl.xticks(fontsize=8)

mpl.yticks(fontsize=8);

mpl.legend().remove();

mpl.savefig("Location Rating.png")

sbn.countplot(zom\_new['rest\_type'])

sbn.countplot(zom\_new['rest\_type']).set\_xticklabels(sbn.countplot(zom\_new['rest\_type']).get\_xticklabels(), rotation=90, ha="right")

fig = mpl.gcf()

fig.set\_size\_inches(12,12)

mpl.savefig("Restaurant Type.png")

mpl.title('Restaurant Type',fontsize=18)

loc\_plt=pds.crosstab(zom\_new['rate'],zom\_new['rest\_type'])

loc\_plt.plot(kind='bar',stacked=True)

mpl.title('Type and rating of the rest.',fontsize=18)

mpl.ylabel('Rest. type',fontsize=12)

mpl.xlabel('Rating',fontsize=12)

mpl.xticks(fontsize=10)

mpl.yticks(fontsize=10)

mpl.legend().remove()

mpl.savefig('Rest Type-Rating.png')

sbn.countplot(zom\_new['type'])

sbn.countplot(zom\_new['type']).set\_xticklabels(sbn.countplot(zom\_new['type']).get\_xticklabels(), rotation=90, ha="right")

fig = mpl.gcf()

fig.set\_size\_inches(14,14)

mpl.title('Service type',fontsize=18)

mpl.savefig('Type of Service.png')

type\_plt=pds.crosstab(zom\_new['rate'],zom\_new['type'])

type\_plt.plot(kind='bar',stacked=True);

mpl.title('Type-Rating Rest.',fontsize=18)

mpl.ylabel('Type',fontsize=12)

mpl.xlabel('Rating',fontsize=12)

mpl.xticks(fontsize=10)

mpl.yticks(fontsize=10);

mpl.savefig('Type and Rating.png')

sbn.countplot(zom\_new['cost'])

sbn.countplot(zom\_new['cost']).set\_xticklabels(sbn.countplot(zom\_new['cost']).get\_xticklabels(), rotation=90, ha="right")

fig = mpl.gcf()

fig.set\_size\_inches(12,12)

mpl.title('Cost(Restaurant)',fontsize=18)

mpl.savefig('Cost(Restaurant).png')